Stream: Internet Engineering Task Force (IETF)
RFC: 9872
Category: Informational
Published: September 2025
ISSN: 2070-1721
Authors: N. Buraglio, *Energy Sciences Network*; T. Jensen; J. Linkova, *Google*

RFC 9872
Recommendations for Discovering IPv6 Prefix Used for IPv6 Address Synthesis

Abstract

On networks providing IPv4-IPv6 translation (RFC 7915), hosts and other endpoints need to know the IPv6 prefix(es) used for translation (the NAT64 prefix (RFC 6052)). This document provides guidelines for NAT64 prefix discovery, specifically recommending obtaining the NAT64 prefix from the Router Advertisement option (RFC 8781) when available.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9872.

Copyright Notice

Table of Contents

1. Introduction

Devices translating between IPv4 and IPv6 packet headers [RFC7915] use a NAT64 prefix to map IPv4 addresses into the IPv6 address space, and vice versa. When a network provides NAT64, it is advantageous for endpoints to acquire the network's NAT64 prefixes (PREF64). Discovering the PREF64 enables endpoints to:

- Implement the customer-side translator (CLAT) function of the 464XLAT architecture [RFC6877].
- Translate IPv4 literals to IPv6 literals (Section 7.1 of [RFC8305]).
- Perform local DNS64 [RFC6147] functions.
- Support applications relying on IPv4 address referral (Section 3.2.2 of [RFC7225]).

Dynamic PREF64 discovery is useful to keep the NAT64 prefix configuration up-to-date, particularly for unmanaged endpoints or endpoints that move between networks. [RFC7050] introduces the first DNS64-based mechanism for PREF64 discovery based on [RFC7051] analysis. However, subsequent methods have been developed to address the [RFC7050] limitations.

For instance, [RFC8781] defines a Neighbor Discovery [RFC4861] option for Router Advertisements (RAs) to convey PREF64 information to hosts. This approach offers several advantages (Section 3 of [RFC8781]), including fate sharing with other host network configuration parameters.

Due to fundamental shortcomings of the [RFC7050] mechanism (Section 4), [RFC8781] is the preferred solution for new deployments. Implementations should strive for consistent PREF64 acquisition methods. The DNS64-based mechanism of [RFC7050] should be employed only when RA-based PREF64 delivery is unavailable or as a fallback for legacy systems incapable of processing the PREF64 RA Option.

2. Terminology

DNS64: A mechanism for synthesizing AAAA records from A records, defined in [RFC6147].

NAT64: A mechanism for translating IPv6 packets to IPv4 packets, and vice versa. The translation is done by translating the packet headers according to the IP/ICMP Translation Algorithm defined in [RFC7915]. NAT64 translators can operate in stateful mode [RFC6144] or stateless mode [RFC6877] (e.g., customer-side translator (CLAT)). This document uses "NAT64" as a generalized term for a translator, which uses the stateless IP/ICMP Translation Algorithm defined in [RFC7915] and operates within a framework for IPv4/IPv6 translation described in [RFC6144].

PREF64 (Pref64::/n or NAT64 prefix): An IPv6 prefix used for IPv6 address synthesis and for translating network addresses and protocols from IPv6 clients to IPv4 servers using the algorithm defined in [RFC6052].

Router Advertisement (RA): A packet used by Neighbor Discovery [RFC4861] and SLAAC to advertise the presence of the routers, together with other IPv6 configuration information.

SLAAC: Stateless Address Autoconfiguration [RFC4862].

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Recommendations for PREF64 Discovery

3.1. Deployment Recommendations for Endpoints

Endpoints **SHOULD** attempt to obtain PREF64 information from RAs per [RFC8781], instead of using the [RFC7050] method. In the absence of the PREF64 information in RAs, an endpoint **MAY** choose to fall back to the mechanism defined in [RFC7050]. This recommendation to prefer the [RFC8781] mechanism over the one defined in [RFC7050] is consistent with Section 5.1 of [RFC8781].

3.2. Deployment Recommendations for Operators

Network operators deploying NAT64 **SHOULD** provide PREF64 information in Router Advertisements per [RFC8781].

3.2.1. Mobile Network Considerations

While [RFC8781] support is widely integrated into modern operating systems on mobile endpoints, equipment deployed in mobile network environments often lacks abilities to include the PREF64 Option into RAs. Therefore, the immediate deployment and enablement of PREF64 by mobile operators may not currently be feasible and the recommendations outlined in this document are not presently applicable to mobile network operators. These environments are encouraged to incorporate [RFC8781] when made practical by infrastructure upgrades or software stack feature additions.

3.2.2. Migration Considerations

Transitioning from the [RFC7050] heuristic to using the [RFC8781] approach might require a period of time where both mechanisms coexist. How long this may take depends on the endpoint footprint, particularly the presence and number of endpoints running outdated operating systems that do not support [RFC8781]. Operators are advised to take those factors into account prior to removing support for the [RFC7050] heuristic, noting that it is still safe to add support for the [RFC8781] approach since endpoints that support it will always prefer it over [RFC7050] if they follow RFC requirements.

Migrating away from DNS64-based discovery also reduces dependency on DNS64 in general, thereby eliminating DNSSEC and DNS64 incompatibility concerns (Section 6.2 of [RFC6147]).

4. Existing Issues with RFC 7050

DNS-based discovery of the NAT64 prefix introduces some challenges, which make this approach less preferable than the latest developed alternatives (such as the PREF64 RA Option [RFC8781]). This section outlines the key issues associated with [RFC7050] with a focus on those not discussed in [RFC7050] or in the analysis of solutions for hosts to discover the NAT64 prefix [RFC7051].

Signalling PREF64 in the RA option addresses all issues outlined in this section (see Section 3 of [RFC8781] for details).

4.1. Dependency on Network-Provided Recursive Resolvers

Fundamentally, the presence of the NAT64 and the exact value of the prefix used for the translation are network-specific attributes. Therefore, [RFC7050] requires the endpoint discovering the prefix to use the DNS64 resolvers provided by the network. If the device or an application is configured to use other recursive resolvers or runs a local recursive resolver, the corresponding name resolution APIs and libraries are required to recognize 'ipv4only.arpa.' as a special name and give it special treatment. This issue and remediation approach are discussed in [RFC8880]. However, it has been observed that very few [RFC7050] implementations support the [RFC8880] requirements for special treatment of 'ipv4only.arpa.'. As a result, configuring such systems and applications to use resolvers other than the one provided by the network breaks the PREF64 discovery, leading to degraded user experience.

VPN applications may override the endpoint's DNS configuration, for example, by configuring enterprise DNS servers as the node's recursive resolvers and forcing all name resolution through the VPN. These enterprise DNS servers typically lack DNS64 functionality and therefore cannot provide information about the PREF64 used within the local network. If the VPN is configured in so-called "split tunneling" mode (when only a subset of network traffic is routed into the VPN tunnel), endpoints may not discover the necessary PREF64, which negatively impacts their connectivity on IPv6-only networks.

If both the network-provided DNS64 and the endpoint's resolver happen to utilize the Well-Known Prefix (64:ff9b::/96) [RFC6052], the endpoint would end up using a PREF64 that's valid for the current network. However, if the endpoint changes its network attachment, it can't detect if the new network lacks NAT64 entirely or uses a network-specific prefix (NSP) [RFC6144] for NAT64.

Signalling PREF64 in an RA option decouples the PREF64 discovery from the host's DNS resolver configuration.

4.2. Network Stack Initialization Delay

When using SLAAC, an IPv6 host typically requires a single RA to acquire its network configuration. For IPv6-only endpoints, timely PREF64 discovery is critical, particularly for those performing local DNS64 or NAT64 functions, such as CLAT [RFC6877]. Until a PREF64 is obtained,

the endpoint's IPv4-only applications and communication to IPv4-only destinations are impaired. The mechanism defined in [RFC7050] does not bundle PREF64 information with other network configuration parameters and requires at least one round-trip time (to send a DNS request and receive a response) after the network stack configuration is completed.

On the other hand, advertising PREF64 in an RA eliminates the period when the host obtains IPv6 addresses and default routers but no PREF64.

4.3. Latency in Updates Propagation

Section 3 of [RFC7050] states:

> The node **SHALL** cache the replies it receives during the Pref64::/n discovery procedure, and it **SHOULD** repeat the discovery process ten seconds before the TTL of the Well-Known Name's synthetic AAAA resource record expires.

As a result, once a PREF64 is discovered, it will be used until the TTL expires or until the node disconnects from the network. There is no mechanism for an operator to force the PREF64 rediscovery on the node without disconnecting the node from the network. If the operator needs to change the PREF64 value used in the network, they need to proactively reduce the TTL value returned by the DNS64 server. This method has two significant drawbacks:

- Many networks utilize external DNS64 servers and therefore have no control over the TTL value if the PREF64 needs to be changed or withdrawn.
- The PREF64 changes need to be planned and executed at least TTL seconds in advance. If the operator needs to notify nodes that a particular prefix must not be used (e.g., during a network outage or if the nodes learned a rogue PREF64 as a result of an attack), it might not be possible without interrupting the network connectivity for the affected nodes.

The mechanism defined in [RFC8781] allows notifying hosts about PREF64 changes immediately by sending an RA with updated information.

4.4. Multihoming Implications

Section 3 of [RFC7050] requires a node to examine all received AAAA resource records to discover one or more PREF64s and to utilize all learned prefixes. However, this approach presents challenges in some multihomed topologies where different DNS64 servers belonging to different ISPs might return different PREF64s. In such cases, it is crucial that traffic destined for synthesized addresses is sent to the correct NAT64 and the source address selected for those flows belongs to the prefix from that ISP's address space. In other words, the node needs to associate each discovered PREF64 with upstream information, including the IPv6 prefix and default gateway. Currently, there is no reliable way for a node to map a DNS64 response (and the prefix learned from it) to a specific upstream in a multihoming scenario. Consequently, the node might inadvertently select an incorrect source address for a given PREF64 and/or send traffic to the incorrect uplink.

Advertising PREF64 in RAs allows hosts to track which PREF64 was advertised by which router and use that information to select the correct next hop. Section 8 of [CLAT] discusses this scenario in more details.

4.5. Security Implications

As discussed in Section 7 of [RFC7050], the DNS-based PREF64 discovery is prone to DNS spoofing attacks. In addition to creating a wider attack surface for IPv6 deployments, [RFC7050] has other security challenges, which are discussed below.

4.5.1. Definition of Secure Channel

[RFC7050] requires a node's communication channel with a DNS64 server to be a "secure channel", which it defines to mean "a communication channel a node has between itself and a DNS64 server protecting DNS protocol-related messages from interception and tampering". This need is redundant when another communication mechanism of IPv6-related configuration, specifically RAs, can already be defended against tampering, for example, by enabling RA-Guard [RFC6105]. Requiring nodes to implement two defense mechanisms when only one is necessary when [RFC8781] is used in place of [RFC7050] creates an unnecessary risk.

4.5.2. Secure Channel Example of IPsec

One of the two examples that [RFC7050] defines to qualify a communication channel with a DNS64 server is the use of an "IPsec-based virtual private network (VPN) tunnel". As of the time of this writing, this is not supported as a practice by any common operating system DNS client. While they could, there have also since been multiple mechanisms defined for performing DNS-specific encryption, such as those defined in [RFC9499], that would be more appropriately scoped to the applicable DNS traffic. These are also compatible with encrypted DNS advertisement by the network using Discovery of Network-designated Resolvers [RFC9463], which would ensure the clients know in advance that the DNS64 server supported the encryption mechanism.

4.5.3. Secure Channel Example of Link Layer Encryption

The other example given by [RFC7050] that would allow a communication channel with a DNS64 server to qualify as a "secure channel" is the use of a "link layer utilizing data encryption technologies". As of the time of this writing, most common link layer implementations use data encryption already with no extra effort needed on the part of network nodes. While this appears to be a trivial way to satisfy this requirement, it also renders the requirement meaningless since any node along the path can still read the higher-layer DNS traffic containing the translation prefix. This seems to be at odds with the definition of "secure channel", as explained in Section 2.2 of [RFC7050].

5. Security Considerations

Obtaining PREF64 information using RAs improves the overall security of an IPv6-only endpoint as it mitigates all attack vectors related to a spoofed or rogue DNS response, as discussed in Section 7 of [RFC7050]. Security considerations related to obtaining PREF64 information from RAs are discussed in Section 7 of [RFC8781].

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <https://www.rfc-editor.org/info/rfc7050>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8781] Colitti, L. and J. Linkova, "Discovering PREF64 in Router Advertisements", RFC 8781, DOI 10.17487/RFC8781, April 2020, <https://www.rfc-editor.org/info/rfc8781>.

7.2. Informative References

[CLAT] Colitti, L., Linkova, J., and T. Jensen, "464XLAT Customer-side Translator (CLAT): Node Recommendations", Work in Progress, Internet-Draft, draft-ietf-v6ops-claton-08, 17 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-v6ops-claton-08>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <https://www.rfc-editor.org/info/rfc6052>.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <https://www.rfc-editor.org/info/rfc6105>.

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <https://www.rfc-editor.org/info/rfc6144>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <https://www.rfc-editor.org/info/rfc6877>.

[RFC7051] Korhonen, J., Ed. and T. Savolainen, Ed., "Analysis of Solution Proposals for Hosts to Learn NAT64 Prefix", RFC 7051, DOI 10.17487/RFC7051, November 2013, <https://www.rfc-editor.org/info/rfc7051>.

[RFC7225] Boucadair, M., "Discovering NAT64 IPv6 Prefixes Using the Port Control Protocol (PCP)", RFC 7225, DOI 10.17487/RFC7225, May 2014, <https://www.rfc-editor.org/info/rfc7225>.

[RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <https://www.rfc-editor.org/info/rfc7915>.

[RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <https://www.rfc-editor.org/info/rfc8305>.

[RFC8880] Cheshire, S. and D. Schinazi, "Special Use Domain Name 'ipv4only.arpa'", RFC 8880, DOI 10.17487/RFC8880, August 2020, <https://www.rfc-editor.org/info/rfc8880>.

[RFC9463] Boucadair, M., Ed., Reddy.K, T., Ed., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", RFC 9463, DOI 10.17487/RFC9463, November 2023, <https://www.rfc-editor.org/info/rfc9463>.

[RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/info/rfc9499>.

Acknowledgments

The authors would like to thank the following people for their valuable contributions: Mike Bishop, Mohamed Boucadair, Lorenzo Colitti, Tom Costello, Charles Eckel, Susan Hares, Nick Heatley, Ted Lemon, Gábor Lencse, David Lou, Peter Schmitt, Éric Vyncke, and Chongfeng Xie.

Authors' Addresses

Nick Buraglio
Energy Sciences Network
Email: buraglio@forwardingplane.net

Tommy Jensen
Email: tojens.ietf@gmail.com

Jen Linkova
Google
Email: furry13@gmail.com